

## (12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro(43) Internationales Veröffentlichungsdatum  
7. Oktober 2004 (07.10.2004)

PCT

(10) Internationale Veröffentlichungsnummer  
WO 2004/086220 A2

(51) Internationale Patentklassifikation<sup>7</sup>: G06F 9/40

(21) Internationales Aktenzeichen: PCT/EP2004/003004

(22) Internationales Anmeldedatum: 22. März 2004 (22.03.2004)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:  
103 13 318.6 25. März 2003 (25.03.2003) DE

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): GIESECKE & DEVRIENT GMBH [DE/DE]; Prinzregentenstrasse 159, 81677 München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder (nur für US): STOCKER, Thomas [DE/DE]; Königseestrasse 48a, 81825 München (DE).

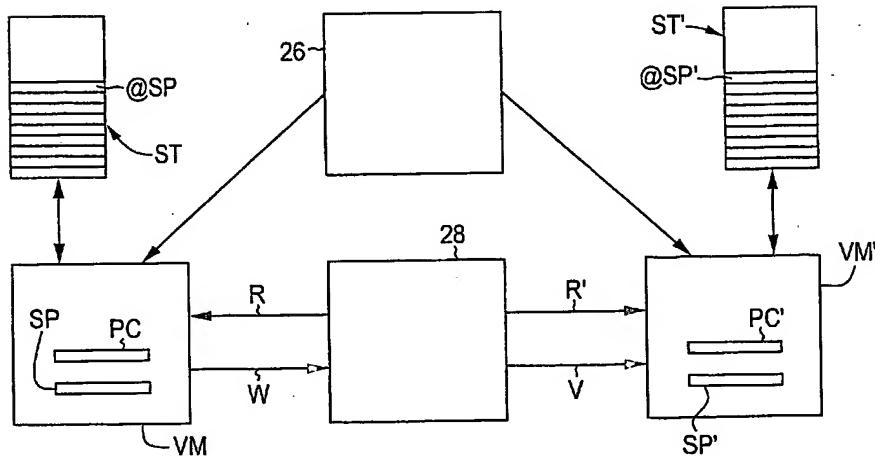
(74) Anwalt: DENDORFER, Claus; Wächtershäuser & Hartz, Weinstrasse 8, 80333 München (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

[Fortsetzung auf der nächsten Seite]

(54) Title: CONTROLLED EXECUTION OF A PROGRAM USED FOR A VIRTUAL MACHINE ON A PORTABLE DATA CARRIER

(54) Bezeichnung: KONTROLIERTE AUSFÜHRUNG EINES FÜR EINE VIRTUELLE MASCHINE VORGESEHENEN PROGRAMMS AUF EINEM TRAGBAREN DATENTRÄGER



(57) Abstract: Disclosed is a method for the controlled execution of a program (26) used for a virtual machine (VM, VM') on a portable data carrier that comprises a processor executing at least one first and a second virtual machine (VM, VM'). According to the inventive method, the program (26) is executed by both the first and the second virtual machine (VM, VM'). Execution of the program is aborted in case a deviation of the mode of operation of the first virtual machine (VM) from the mode of operation of the second virtual machine (VM') is detected during execution of the program (26). A data carrier and a computer program are provided with corresponding characteristics. The invention creates technology for controlled program execution, which prevents safety hazards due to an attack or malfunction of the data carrier.

[Fortsetzung auf der nächsten Seite]

WO 2004/086220 A2



(84) **Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart):** ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO Patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

**Erklärung gemäß Regel 4.17:**

— *hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii) für die folgenden Bestimmungsstaaten AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU,*

**Veröffentlicht:**

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

*Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.*

(57) **Zusammenfassung:** Bei einem Verfahren zur kontrollierten Ausführung eines für eine virtuelle Maschine (VM, VM') vorgesehenen Programms (26) auf einem tragbaren Datenträger, wobei der Datenträger einen Prozessor aufweist, der mindestens eine erste und eine zweite virtuelle Maschine (VM, VM') ausführt, wird das Programm (26) sowohl von der ersten als auch von der zweiten virtuellen Maschine (VM, VM') ausgeführt. Falls während der Ausführung des Programms (26) eine Abweichung des Betriebszustands der ersten virtuellen Maschine (VM) von dem Betriebszustand der zweiten virtuellen Maschine (VM') festgestellt wird, wird die Programmausführung abgebrochen. Ein Datenträger und ein Computerprogrammprodukt weisen entsprechende Merkmale auf. Die Erfindung stellt eine Technik zur kontrollierten Programmausführung bereit, die Sicherheitsrisiken durch einen Angriff oder eine Betriebsstörung des Datenträgers vermeidet.

Kontrollierte Ausführung eines für eine virtuelle Maschine  
vorgesehenen Programms auf einem tragbaren Datenträger

5 Die Erfindung betrifft allgemein das technische Gebiet der Ausführung eines für eine virtuelle Maschine vorgesehenen Programms auf einem tragbaren Datenträger, der einen Prozessor aufweist. Ein derartiger tragbarer Datenträger kann insbesondere eine Chipkarte in unterschiedlichen Bauformen oder ein Chipmodul sein. Spezieller betrifft die Erfindung die kontrollierte

10 Programmausführung, um Störungen oder Angriffe zu erkennen und um zu verhindern, daß die Sicherheit des tragbaren Datenträgers durch solche Störungen oder Angriffe kompromittiert wird.

Tragbare Datenträger, die eine virtuelle Maschine zur Ausführung von

15 Programmen aufweisen, sind z.B. unter der Marke *Java Card*™ bekannt. Solche Datenträger sind in Kapitel 5.10.2 des Buches "Handbuch der Chipkarten" von W. Rankl und W. Effing, Hanser Verlag, 3. Auflage, 1999, Seiten 261 - 281, beschrieben. Eine ausführliche Spezifikation des *Java-Card-Standards*, der dabei verwendeten virtuellen Maschine *JCVM (Java Card Virtual Machine)* und der ausgeführten Programme (*Java Card Applets*) findet sich auf den Internet-Seiten der Firma Sun Microsystems, Inc., unter <http://java.sun.com/products/javacard>.

Tragbare Datenträger werden häufig für sicherheitskritische Anwendungen

25 eingesetzt, beispielsweise im Zusammenhang mit Finanztransaktionen oder bei der elektronischen Signatur von Dokumenten. Es sind bereits Techniken zum Angriff auf tragbare Datenträger bekannt geworden, bei denen die Programmausführung durch externe Beeinflussung gestört wird. Eine solche Störung kann insbesondere durch Spannungsimpulse, Einwirkung von

30 Wärme oder Kälte, elektrische oder magnetische Felder, elektromagnetische Wellen oder Teilchenstrahlung verursacht werden. So ist es z.B. möglich,

durch Lichtblitze auf den freigelegten Halbleiterchip des tragbaren Datenträgers Registerinhalte im Prozessor oder Speicherinhalte zu verändern. Durch eine derartige Störung kann möglicherweise die Sicherheit des Datenträgers kompromittiert werden, wenn z.B. der Datenträger einen fehlerhaft ver-  
5 schlüsselten Text ausgibt, dessen Analyse Rückschlüsse auf einen geheimen Schlüssel zuläßt.

Es besteht daher das Problem, einen Datenträger der eingangs genannten Art gegen eine Kompromittierung durch Angriffe abzusichern, bei denen die  
10 Ausführung eines Programms durch eine virtuelle Maschine gestört wird.

Aus GB 2 353 113 A ist ein Computernetz bekannt, das Softwarefehler in gewissem Umfang auszugleichen vermag. Bei diesem Computernetz sind mindestens zwei Rechner vorgesehen, die je eine virtuelle Maschine ausführen.  
15 Wenn ein fehlerhafter Ablauf einer der virtuellen Maschinen festgestellt wird, wird die Programmausführung durch die andere virtuelle Maschine bzw. die anderen virtuellen Maschinen fortgesetzt.

Das aus GB 2 353 113 A bekannte System ist gattungsfremd, da es nicht für  
20 einen tragbaren Datenträger, sondern für ein komplexes Netz mit mehreren Computern vorgesehen ist. Die virtuellen Maschinen werden von mehreren Prozessoren ausgeführt, die nur lose aneinander gekoppelt sind. Die Programmausführung wird auch bei einer Störung einer virtuellen Maschine fortgesetzt. Zur Anwendung bei einem tragbaren Datenträger mit einem  
25 einzigen Prozessor eignet sich diese Lehre nicht.

Die Erfindung hat demgemäß die Aufgabe, die Probleme des Standes der Technik zumindest zum Teil zu vermeiden und eine Technik zur kontrollierten Ausführung eines für eine virtuelle Maschine vorgesehenen Programms

auf einem tragbaren Datenträger bereitzustellen, durch die Sicherheitsrisiken bei einem Angriff oder einer Betriebsstörung vermieden werden. In bevorzugten Ausgestaltungen soll ein zuverlässiger Schutz bei möglichst geringen Leistungseinbußen des tragbaren Datenträgers erreicht werden.

5

Erfindungsgemäß wird diese Aufgabe ganz oder zum Teil gelöst durch ein Verfahren mit den Merkmalen des Anspruchs 1, einen tragbaren Datenträger mit den Merkmalen des Anspruchs 10 und ein Computerprogrammprodukt mit den Merkmalen des Anspruchs 11. Die abhängigen Ansprüche betreffen bevorzugte Ausgestaltungen der Erfindung.

10

Die Erfindung beruht auf der Grundidee, von dem einen Prozessor des tragbaren Datenträgers mehrere virtuelle Maschinen ausführen zu lassen, die ihrerseits ein und dasselbe Programm ausführen. Durch diese Maßnahme ergibt sich eine Redundanz bei der Programmausführung, die zur Erkennung von Fehlfunktionen eingesetzt werden kann. Es ist ein überraschendes Ergebnis der vorliegenden Erfindung, daß eine derartige Redundanz auch bei einem tragbaren Datenträger mit nur einem einzigen Prozessor erzielt werden kann.

15

Erfindungsgemäß wird die Ausführung des Programms abgebrochen, falls eine Abweichung der Betriebszustände der virtuellen Maschinen voneinander festgestellt wird. Es wird also nicht versucht, eine der virtuellen Maschinen als korrekt arbeitend zu identifizieren und die Programmausführung mit dieser virtuellen Maschine fortzusetzen. Durch den erfindungsgemäß vorgesehenen Programmabbruch wird eine besonders hohe Angriffssicherheit erreicht.

20

25

- 4 -

Die Erfindung bietet den erheblichen Vorteil, daß sie problemlos auf üblicher Hardware implementiert werden kann. Überdies ist keinerlei Anpassung des auszuführenden Programms an den erfindungsgemäßen Angriffsschutz erforderlich. Alle für die standardgemäße virtuelle Maschine vorgesehenen  
5 Programme laufen unverändert auf den erfindungsgemäß ausgestalteten Datenträgern, was die Akzeptanz der Erfindung sehr fördert.

Bei der Überprüfung der Betriebszustände der virtuellen Maschinen auf Übereinstimmung findet vorzugsweise kein vollständiger Vergleich statt.  
10 Vielmehr ist in bevorzugten Ausführungsformen lediglich ein Vergleich von Inhalten einiger wichtiger Register und/oder Speicherinhalten vorgesehen. Solche wichtigen Register können beispielsweise die Programmzähler und/oder die Stapelzeiger der virtuellen Maschinen sein. Als Beispiel für wichtige Speicherinhalte, die in manchen Ausgestaltungen der Erfindung  
15 miteinander verglichen werden, ist das jeweils jüngste ("oberste") Element in den Stapspeichern der virtuellen Maschinen zu nennen.

Da die virtuellen Maschinen auf dem tragbaren Datenträger von einem einzigen Prozessor ausgeführt werden, erfolgt in der Regel ein ineinander verzahnter (*interleaved*) Programmablauf. Dies schließt jedoch nicht aus, daß einzelne Operationen echt parallel ausgeführt werden, wenn der Prozessor des tragbaren Datenträgers dazu eingerichtet ist.  
20

Die Überprüfung der Betriebszustände der virtuellen Maschinen kann zu beliebigen Zeitpunkten erfolgen, an denen die virtuellen Maschinen – bei fehlerfreiem Ablauf – identische Zustände aufweisen müßten. Obwohl die Überprüfung also prinzipiell nicht an die Befehlsgrenzen des ausgeführten Programms gebunden ist, ist in bevorzugten Ausführungsformen vorgesehen, diese Überprüfung nach jeder Ausführung eines Befehls des Pro-  
25

- 5 -

gramms durch die virtuellen Maschinen vorzunehmen. In alternativen Ausgestaltungen kann der Abgleich der virtuellen Maschinen entweder schon nach der Ausführung von Teilen von Befehlen oder jeweils erst nach der Ausführung mehrerer Befehle erfolgen.

5

Vorzugsweise wird jeder Befehl des Programms zunächst von der ersten und dann von der zweiten virtuellen Maschine ausgeführt. In manchen Ausgestaltungen wird hierbei zunächst die Ausführung des Befehls von der ersten virtuellen Maschine beendet, bevor der Prozessor des tragbaren

10 Datenträgers mit der Ausführung des Befehls durch die zweite virtuelle Maschine beginnt. In anderen Ausgestaltungen kann der Prozessor dagegen mehrere Teilabschnitte des Befehls jeweils zunächst auf der ersten und dann auf der zweiten virtuellen Maschine ausführen, sofern nur die erste virtuelle Maschine nicht gegenüber der zweiten virtuellen Maschine ins Hintertreffen  
15 gerät.

Durch den Einsatz von mindestens zwei - in manchen Ausführungsformen auch mehr - virtuellen Maschinen verringert sich die für jede virtuelle Maschine bereitstehende Rechenleistung entsprechend. Hinsichtlich der  
20 tatsächlichen Programmlaufzeit ist jedoch zu berücksichtigen, daß bei einem typischen tragbaren Datenträger viel Zeit für Schreibvorgänge in einen nicht-flüchtigen Speicher des Datenträgers benötigt wird.

In bevorzugten Ausführungsformen ist deshalb vorgesehen, daß die virtuellen Maschinen auf einen gemeinsamen Heap (Haufen oder Halde) im nicht-flüchtigen Speicher des tragbaren Datenträgers zugreifen, wobei Schreibvorgänge nur von einer der virtuellen Maschinen ausgeführt werden. Die andere virtuelle Maschine oder die anderen virtuellen Maschinen kann/  
25 können entweder den Schreibvorgang ganz überspringen oder statt des

Schreibvorgangs überprüfen, ob an der zu beschreibenden Stelle im Speicher bereits der einzuschreibende Wert enthalten ist. Falls das auszuführende Programm viele Schreibvorgänge in den Heap enthält, benötigen diese einen wesentlichen Teil der Gesamtausführungszeit. Durch den Einsatz der gerade beschriebenen Ausführungsform der Erfindung bleibt dieser Teil der Gesamtausführungszeit unverändert, während nur die reine Rechenzeit des Prozessors – die, wie erwähnt, weniger ins Gewicht fällt – ansteigt.

Der erfindungsgemäße tragbare Datenträger ist vorzugsweise als Chipkarte oder Chipmodul ausgebildet. Das erfindungsgemäße Computerprogrammprodukt weist Programmbefehle auf, um das erfindungsgemäße Verfahren zu implementieren. Ein derartiges Computerprogrammprodukt kann ein körperliches Medium sein, beispielsweise ein Halbleiterspeicher oder eine Diskette oder eine CD-ROM, auf dem ein Programm zur Ausführung eines erfindungsgemäßen Verfahrens gespeichert ist. Das Computerprogrammprodukt kann jedoch auch ein nicht-körperliches Medium sein, beispielsweise ein über ein Computernetzwerk übermitteltes Signal. Das Computerprogrammprodukt kann insbesondere zur Verwendung im Zusammenhang mit der Herstellung und/oder Initialisierung und/oder Personalisierung von Chipkarten oder sonstigen Datenträgern vorgesehen sein.

In bevorzugten Ausgestaltungen weisen der Datenträger und/oder das Computerprogrammprodukt Merkmale auf, die den oben beschriebenen und/oder den in den abhängigen Verfahrensansprüchen genannten Merkmälen entsprechen.

Weitere Merkmale, Vorteile und Aufgaben der Erfindung gehen aus der folgenden genauen Beschreibung eines Ausführungsbeispiels und mehrerer

- 7 -

Ausführungsalternativen hervor. Es wird auf die schematischen Zeichnungen verwiesen, in denen zeigen:

Fig. 1 ein Blockdiagramm mit Funktionseinheiten eines tragbaren Daten-  
trägers nach einem Ausführungsbeispiel der Erfindung,

Fig. 2 eine konzeptuelle Darstellung von Komponenten, die bei der Pro-  
grammausführung durch den tragbaren Datenträger aktiv sind,

10 Fig. 3 ein Flußdiagramm einer während der Programmausführung für jeden  
Programmbefehl durchlaufenen Hauptschleife, und

Fig. 4 ein Flußdiagramm der Überprüfung der Betriebszustände der virtu-  
ellen Maschinen auf Übereinstimmung.

15

Der in Fig. 1 dargestellte Datenträger 10 ist im vorliegenden Ausführungs-  
beispiel als Chipkarte gemäß dem *Java-Card*-Standard ausgestaltet. Der Da-  
tenträger weist auf einem einzigen Halbleiterchip einen Prozessor 12, meh-  
rere in unterschiedlichen Technologien ausgestaltete Speicherfelder und eine  
20 Schmittstellenschaltung 14 zur kontaktlosen oder kontaktgebundenen Kom-  
munikation auf. Im vorliegenden Ausführungsbeispiel sind als Speicherfel-  
der ein Arbeitsspeicher 16, ein Festwertspeicher 18 und ein nicht-flüchtiger  
Speicher 20 vorgesehen. Der Arbeitsspeicher 16 ist als RAM, der Festwert-  
speicher 18 als maskenprogrammiertes ROM und der nicht-flüchtige Spei-  
cher 20 als elektrisch lösch- und programmierbares EEPROM ausgestaltet.  
25 Schreibzugriffe auf den nicht-flüchtigen Speicher 20 sind relativ aufwendig  
und benötigen beispielsweise die dreifache Zeit eines Lesezugriffs.

Im Festwertspeicher 18 – und zum Teil auch im nicht-flüchtigen Speicher 20 – ist ein Betriebssystem 22 enthalten, das eine Vielzahl von Funktionen und Diensten bereitstellt. Unter anderem weist das Betriebssystem 22 ein Code-  
modul 24 auf, das eine virtuelle Maschine – im vorliegenden Ausführungs-  
5 beispiel eine JCVM (*Java Card Virtual Machine*) – implementiert.

In Fig. 1 ist beispielhaft ein auszuführendes Programm 26 im nicht-flüchtigen Speicher 20 gezeigt, das im vorliegenden Ausführungsbeispiel als *Java Card Applet* ausgestaltet ist. Das Programm 26 kann auch ganz oder teilweise  
10 im Festwertspeicher 18 enthalten sein, und es können weitere Programme zur Ausführung durch den tragbaren Datenträger 10 vorgesehen sein. Ein Bereich im nicht-flüchtigen Speicher 20 ist als Heap 28 (Haufen oder Halde) reserviert, um während der Programmausführung Objekte und andere Datenstrukturen aufzunehmen.

15 Um das Programm 26 auszuführen, startet der Prozessor 12 unter Steuerung des Betriebssystems 22 zwei Instanzen des Codemoduls 24, die je eine virtuelle Maschine VM, VM' bilden. Wie in Fig. 2 dargestellt ist, führen die beiden virtuellen Maschinen VM, VM' ein und dasselbe Programm 26 aus, das im nicht-flüchtigen Speicher 20 nur einmal vorliegt. Die beiden virtuellen Maschinen VM, VM' greifen daher, wenn sie einen Befehl des Programms 26  
20 holen, auf identische Adressen im nicht-flüchtigen Speicher 20 zu.

25 Die beiden virtuellen Maschinen VM, VM' führen während des Programmablaufs Zugriffsoperationen auf den gemeinsamen Heap 28 aus. Auch hier sind die im Heap 28 gespeicherten Objekte und Datenstrukturen jeweils nur einmal vorhanden. Die erste virtuelle Maschine VM führt auf dem Heap 28 sowohl Leseoperationen R als auch Schreiboperationen W aus. Die zweite

- 9 -

virtuelle Maschine VM' führt dagegen zwar Leseoperationen R' aus, aber keine Schreiboperationen, sondern Verifikationsoperationen V.

Die virtuellen Maschinen VM, VM' weisen jeweils eigene Register auf, von 5 denen in Fig. 2 je ein Programmzähler PC, PC' und ein Stapelzeiger SP, SP' gezeigt sind. Diese Register sind im Arbeitsspeicher 16 angelegt oder werden durch Register des Prozessors 12 implementiert. Ferner weist jede virtuelle Maschine VM, VM' einen eigenen Stapspeicher ST, ST' auf, der in je einem Bereich des Arbeitsspeichers 16 angelegt ist. Die jeweils jüngsten ("obersten") 10 Einträge in den Stapspeichern ST, ST', auf die die jeweiligen Stapelzeiger SP, SP' verweisen, sind in Fig. 2 durch @SP und @SP' gekennzeichnet.

In einer Abwandlung zu der in Fig. 1 gezeigten Darstellung können die virtuellen Maschinen VM, VM' hardwaremäßig getrennt in separaten 15 Speichern 20 angelegt sein, die weiterhin separaten Prozessoren zugeordnet sein können. Vorgesehen sein kann auch, daß die virtuellen Maschinen VM, VM' als Hardwarebauelemente ausgeführt sind.

Bei der Ausführung des Programms 26 führt das Betriebssystem 22 die in 20 Fig. 3 gezeigte Schleife aus. Für jeden Befehl des Programms 26 erfolgt ein Schleifendurchlauf. Der Befehl wird zunächst in Schritt 30 von der ersten virtuellen Maschine VM ausgeführt. Hierbei bestehen im Vergleich zu der Programmausführung bei einem System nach dem Stand der Technik durch eine einzige virtuelle Maschine keine Unterschiede. Insbesondere verwaltet 25 die erste virtuelle Maschine ihre Register PC und SP sowie den Stapspeicher ST und führt gegebenenfalls eine Leseoperation R und/oder eine Schreiboperation W auf dem Heap 28 durch.

Wenn die Befehlsausführung durch die erste virtuelle Maschine VM abgeschlossen ist, führt die zweite virtuelle Maschine VM' in Schritt 32 denselben Befehl des Programms 26 nochmals aus. Auch hier erfolgen die Verwaltung der Register PC' und SP', die Verwaltung des Stapelspeichers ST' sowie 5 gegebenenfalls eine Operation R' des Lesens vom Heap 28 auf übliche Weise.

Die Befehlsausführung durch die zweite virtuelle Maschine VM' unterscheidet sich jedoch von der Befehlsausführung durch die erste virtuelle Maschine VM dadurch, daß statt einer durch den Befehl gegebenenfalls vorgegebenen Schreiboperation eine Vergleichsoperation V ausgeführt wird, bei der 10 der eigentlich in den Heap 28 zu schreibende Wert mit dem aktuellen Inhalt des Heaps 28 an der zu beschreibenden Adresse verglichen wird. Wenn die Berechnungsvorgänge der beiden virtuellen Maschinen VM, VM' übereinstimmen, dann hat die erste virtuelle Maschine VM in Schritt 30 den nun in 15 Schritt 32 von der zweiten virtuellen Maschine VM' bestimmten Wert bereits in den Heap 28 eingeschrieben. Die Verifikationsoperation V in Schritt 32 ergibt daher eine Übereinstimmung, und der Berechnungsablauf wird fortgesetzt. Wird in Schritt 32 dagegen eine Abweichung der Werte festgestellt, so deutet dies auf eine Fehlfunktion einer der virtuellen Maschinen VM, VM' 20 hin. Die Programmausführung wird dann als fehlerhaft abgebrochen. Diese Möglichkeit ist in Fig. 3 durch einen gestrichelten Pfeil angedeutet.

Nach der Ausführung des Programmbeehls durch beide virtuellen Maschinen VM, VM' wird nun in Schritt 34 geprüft, ob die erreichten Betriebszustände 25 der beiden virtuellen Maschinen übereinstimmen. Dazu werden im hier beschriebenen Ausführungsbeispiel nur einige Register- und Speicherwerte auf Übereinstimmung geprüft, wie dies in Fig. 4 gezeigt ist. Zunächst wird in Teilschritt 34.1 überprüft, ob die beiden Programmzähler PC, PC' nach der Befehlsausführung denselben Wert aufweisen. Ist dies der Fall, so

erfolgt in Teilschritt 34.2 eine Überprüfung der beiden Stapelzeiger SP, SP' auf Übereinstimmung. Wenn auch dieser Test erfolgreich ist, wird in Teilschritt 34.3 überprüft, ob die jeweils jüngsten Einträge @SP, @SP' in den Stapelspeichern ST, ST', also die Einträge, auf die die Stapelzeiger SP, SP' 5 verweisen, identisch sind.

Wenn bei allen drei Abfragen 34.1, 34.2, 34.3 eine Übereinstimmung festgestellt worden ist, so wird in Schritt 34 (Fig. 3) von einer korrekten Programmausführung durch die beiden virtuellen Maschinen VM, VM' ausgegangen. Es erfolgt dann ein Rücksprung zum Anfang der Schleife, und der 10 nächste Befehl des Programms 26 wird zunächst durch die erste und dann durch die zweite virtuelle Maschine VM, VM' ausgeführt.

Ergibt sich bei der Zustandsüberprüfung in einem der drei Teilschritte 34.1, 15 34.2, 34.3 jedoch eine Abweichung, so deutet dies auf eine Fehlfunktion einer der beiden virtuellen Maschinen VM, VM' hin. Dies wiederum wird als Indiz für eine Störung oder einen Angriff auf die Hardware des tragbaren Datenträgers 10 erachtet. Da der Prozessor 12 die Schritte 30 und 32 streng nacheinander ausführt, ist bei einem Angriff z.B. durch einen Lichtblitz nur die 20 Funktion einer der beiden virtuellen Maschinen VM, VM' betroffen. Selbst bei rasch aufeinanderfolgenden Lichtblitzen wäre es unwahrscheinlich, daß beide virtuellen Maschinen VM, VM' in gleicher Weise gestört werden würden.

25 Wird in Schritt 34 eine Abweichung der Betriebszustände festgestellt, so wird die Programmausführung als fehlerhaft abgebrochen. Das Betriebssystem 22 bringt dann den Datenträger 10 in einen sicheren Zustand. Hierbei ist insbesondere zu beachten, daß der Datenträger 10 nach einem Programmabbruch keine Ausgabeoperationen mehr durchführen soll. Je nach den an

- 12 -

den Datenträger 10 gestellten Sicherheitsanforderungen kann vorgesehen sein, daß der Datenträger 10 nach einem regulären Rücksetzvorgang (*reset*) wieder einsatzbereit ist, oder es kann ein besonderer Freischaltvorgang gefordert werden, oder der Datenträger 10 kann gänzlich deaktiviert werden.

5

Patentansprüche

1. Verfahren zur kontrollierten Ausführung eines für eine virtuelle  
5 Maschine (VM, VM') vorgesehenen Programms (26) auf einem  
tragbaren Datenträger (10), wobei
  - der Datenträger (10) einen Prozessor (12) aufweist, der minde-  
stens eine erste und eine zweite virtuelle Maschine (VM, VM')  
ausführt,
  - das Programm (26) sowohl von der ersten als auch von der zweien  
10 virtuellen Maschine (VM, VM') ausgeführt wird,
  - der Betriebszustand der ersten virtuellen Maschine (VM) und der  
Betriebszustand der zweiten virtuellen Maschine (VM') während  
der Ausführung des Programms (26) auf Übereinstimmung über-  
prüft werden, und
  - die Ausführung des Programms (26) abgebrochen wird, falls eine  
15 Abweichung des Betriebszustands der ersten virtuellen Maschine  
(VM) von dem Betriebszustand der zweiten virtuellen Maschine  
(VM') festgestellt wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die  
Überprüfung des Betriebszustands der ersten virtuellen Maschine  
(VM) und des Betriebszustand der zweiten virtuellen Maschine  
(VM') auf Übereinstimmung eine Überprüfung umfaßt, ob der  
Stand eines Programmzählers (PC) der ersten virtuellen Maschine  
25 (VM) gleich dem Stand eines Programmzählers (PC') der zweiten  
virtuellen Maschine (VM') ist.
3. Verfahren nach Anspruch 1 oder Anspruch 2, dadurch gekenn-  
30 zeichnet, daß die Überprüfung des Betriebszustands der ersten  
virtuellen Maschine (VM) und des Betriebszustand der zweiten

virtuellen Maschine (VM') auf Übereinstimmung eine Überprüfung umfaßt, ob der Stand eines Stapelzeigers (SP) der ersten virtuellen Maschine (VM) gleich dem Stand eines Stapelzeigers (SP') der zweiten virtuellen Maschine (VM') ist.

5

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß die Überprüfung des Betriebszustands der ersten virtuellen Maschine (VM) und des Betriebszustand der zweiten virtuellen Maschine (VM') auf Übereinstimmung eine Überprüfung umfaßt, ob der Wert des jüngsten Elements (@SP) in einem der ersten virtuellen Maschine (VM) zugeordneten Stapspeicher (ST) gleich dem Wert des jüngsten Elements (@SP') in einem der zweiten virtuellen Maschine (VM') zugeordneten Stapspeicher (ST') ist.
- 10 15 20 25 5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß die Überprüfung des Betriebszustands der ersten virtuellen Maschine (VM) und des Betriebszustand der zweiten virtuellen Maschine (VM') auf Übereinstimmung jeweils vorgenommen wird, nachdem ein Befehl des Programms (26) sowohl von der ersten als auch von der zweiten virtuellen Maschine (VM, VM') ausgeführt worden ist.
6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß die erste und die zweite virtuelle Maschine (VM, VM') auf einen gemeinsamen Heap (28) in einem nicht-flüchtigen Speicher (20) des Datenträgers (10) zugreifen.

5

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß bei der Ausführung eines Befehls des Programms (26), der einen Schreibvorgang auf den gemeinsamen Heap (28) beinhaltet, der Schreibvorgang nur von der ersten virtuellen Maschine (VM) ausgeführt wird.
8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, daß der Befehl des Programms (26) zuerst von der ersten virtuellen Maschine (VM) und dann von der zweiten virtuellen Maschine (VM<sup>1</sup>) ausgeführt wird, und daß die zweite virtuelle Maschine (VM<sup>1</sup>) statt des Schreibvorgangs überprüft, ob im Heap (28) an der zu beschreibenden Stelle der einzuschreibende Wert enthalten ist.
9. Verfahren nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, daß das Programm (26) ein *Java Card Applet* ist, das zur Ausführung durch eine JCVM (*Java Card Virtual Machine*) vorgesehen ist.
10. Tragbarer Datenträger (10), insbesondere Chipkarte oder Chipmodul, mit einem Prozessor (12) und einem Betriebssystem (22), wobei das Betriebssystem (22) Programmbefehle aufweist, um den Prozessor (12) zur Ausführung eines Verfahrens nach einem der Ansprüche 1 bis 9 zu veranlassen.
11. Computerprogrammprodukt, das Programmbefehle aufweist, um einen Prozessor (12) eines tragbaren Datenträgers (10) zu veranlassen, ein Verfahren mit den Merkmalen eines der Ansprüche 1 bis 9 auszuführen.

20

15

25

1/2

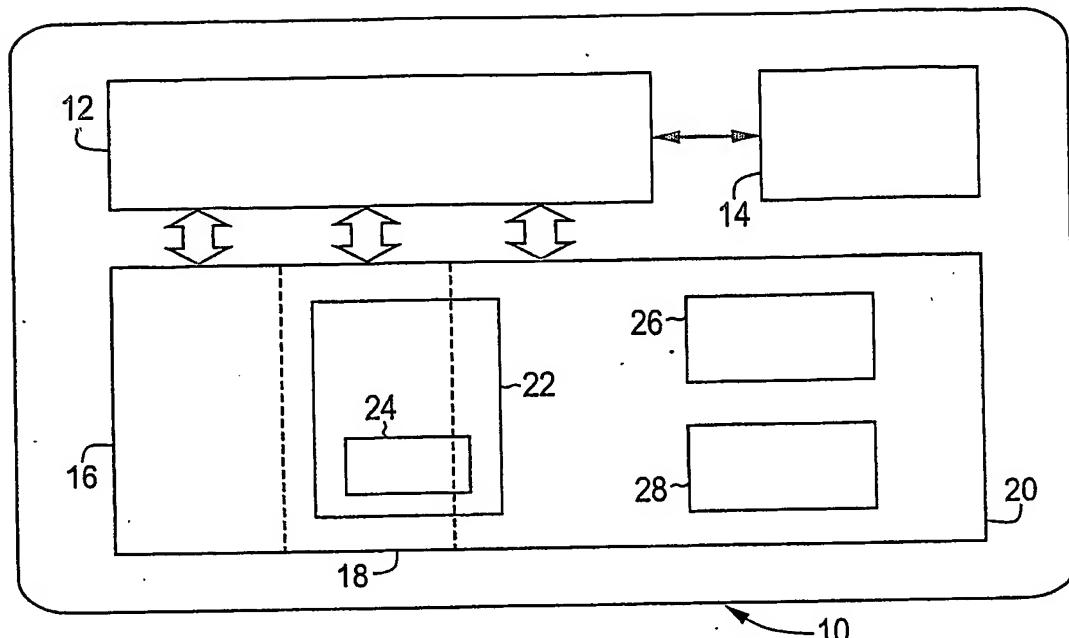


Fig. 1

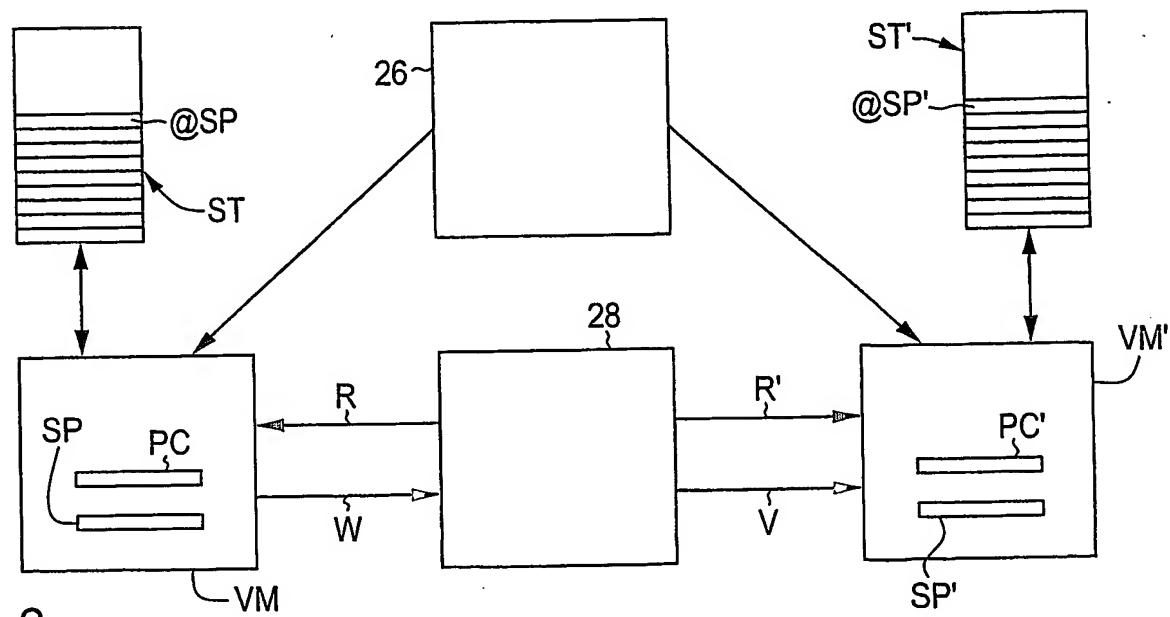


Fig. 2

2/2

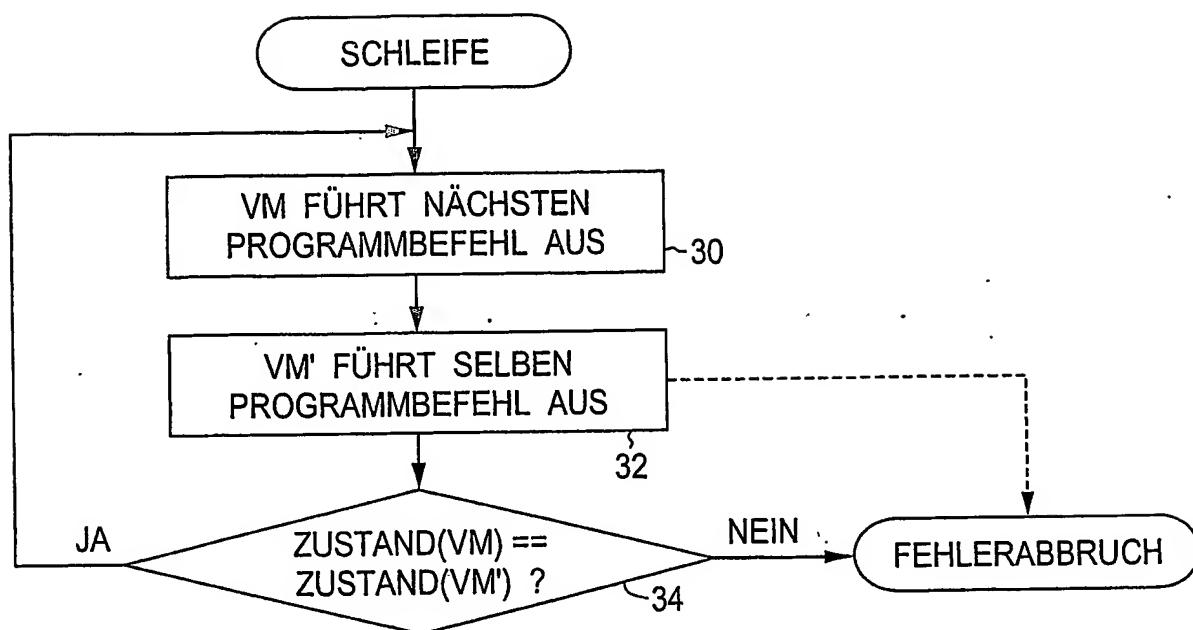


Fig. 3

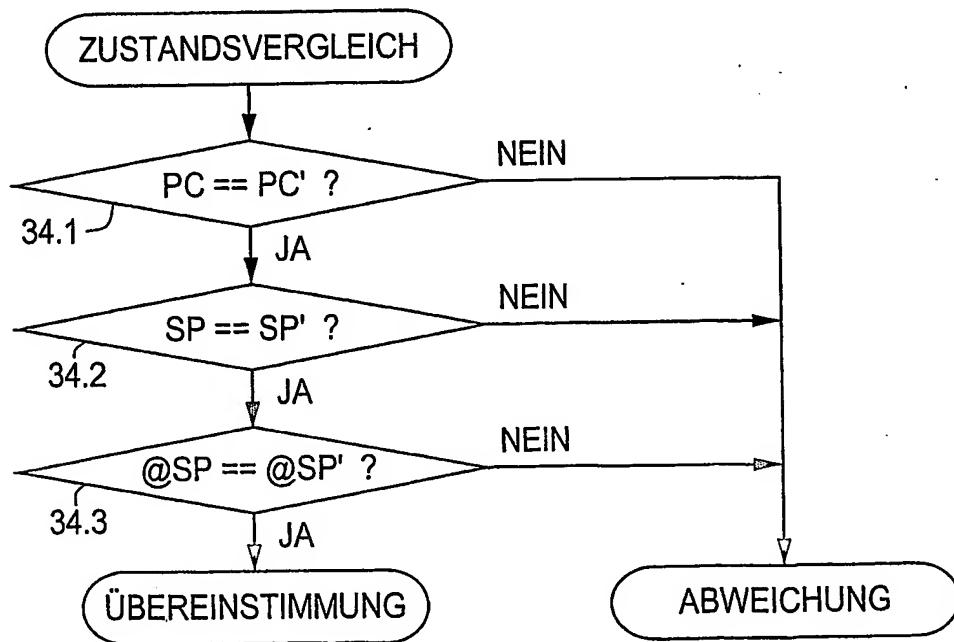


Fig. 4